**1 OPC**的基础知识

**1.1 OPC**的开发背景和历史

**1.1.1** 为什么需要**OPC**

对于早期的计算机系统，为了实现不同的硬件和软件所构成的计算机之间的数据 交换和通信，必须要花费很多时间去开发独自的通信程序。但是正是由于现在有了数 据交换和通信的工业标准，才有可以实现象互联网那样，使不同的计算机相互连接的 巨大网络。所以在开发企业的信息系统时，采用符合工业标准的数据库和客户－服务 器接口，可以使有效的精力更多地投入到应用程序本身功能的开发中去。

工业制造系统也存在同样的问题。也就是使由不同的供应商提供的机器设备无须 特别的软件开发就可以互相连接。例如在实现象图 1-1那样的多层生产控制信息系统 时，从处理设备数据的现场设备层，到进行过程处理的过程控制系统层，以至最上层 的生产管理层，建立和普及一个有效的数据交换工业标准乃是当务之急。在这种情况 下，利用微软Windows视窗中的OLE/COM技术实现工业制造系统过程控制中的数据交 换标准化，正是OPC（OLE for Process Control）本来的目的所在。

应用程序

生产管理层

操作平台

历史数据 服务器

操作平台 历史数据

服务器

控制器

控制器

过程控制层

现场总线

传感器

-压力

-温度

-流量

-液位

阀门 定位器



PD 仪表 分析仪

现场设备层

-模拟 I/O

-数字 I/O

-TC/RTD

图 *1-1* 生产控制信息系统的构成

**1.1.2 OPC**怎样解决你的问题

到目前为止，硬件的驱动器和与其连接的应用程序之间的接口并没有统一的标准。 例如，在FA（Factory Automation）领域，连接PLC（Programmable Logic Controller）等 控制设备和 SCADA/HMI软件，需要不同的FA网络系统构成。根据某调查结果，据说 在控制系统软件开发的所需费用中，各种各样机器的应用程序设计占费用的７成，而 开发机器设备间的连接接口则占了３成。此外，在PA（Process Automation）领域，当 希望把DCS（Distributed Control System）中所有的过程数据传送到生产管理系统时，

必须按照各个供应厂商的各个机种开发特定的接口（例如，利用C语言DLL连接的DDE

服务器或者利用FTP的文本文件传送等）。

例如，在图 1-2所示的由４种控制设备和与其连接的监视，趋势图以及表报３种 应用程序所构成的系统时，必须花费大量时间去开发分别对应设备A，B，C，D的监 视，趋势图以及表报应用程序的接口软件共计１２种驱动器。同时由于系统中共存各 种各样的驱动器，也使维护运转环境的稳定性和信赖性更加困难。

监视 应用程序

趋势图 应用程序

报表 应用程序

驱动器A 驱动器B 驱动器C 驱动器D

驱动器A

驱动器B

驱动器C

驱动器D

监视 应用程序

趋势图 应用程序

报表 应用程序

设备A

设备D

设备B

设备C

设备A

设备B 设备C

设备D

图 *1-2* 利用驱动器的系统连接

OPC是为了不同供应厂商的设备和应用程序之间的软件接口标准化，使其间的数 据交换更加简单化的目的而提出的。作为结果，从而可以向用户提供不依靠于特定开 发语言和开发环境的可以自由组合使用的过程控制软件组件产品。

利用OPC的系统，是由按照应用程序（客户程序）的要求提供数据收集服务的OPC 服务器，使用OPC服务器所必需的OPC接口，以及接受服务的OPC应用程序所构成。 OPC服务器是按照各个供应厂商的硬件所开发的，使之可以吸收各个供应厂商硬件和 系统的差异，从而实现不依存于硬件的系统构成。同时利用一种叫做Variant的数据型， 可以不依存于硬件中固有数据型，按照应用程序要求提供的数据格式。

利用OPC使接口标准化可以构成如图 1-3所示的系统。用户可以不依存于设备A，

B，C，D的内部结构及它的供应厂商，选用监视，趋势图以及表报应用程序。

监视 应用程序

OPC

监视 应用程序

OPC

趋势图 应用程序

OPC

趋势图 应用程序

OPC

报表 应用程序

OPC

报表 应用程序

OPC

OPC OPC OPC OPC

OPC

OPC

OPC

OPC

设备 A

设备D

设备 A

设备B 设备C

图 *1-3* 利用*OPC*的控制系统构成

设备B

设备C

设备D

**1.1.3 OPC**的历史

早期的OPC标准是由提供工业制造软件的５家公司所组成的OPC特别工作小组所 开发的。Fisher-Rosement，Intellution，Rockwell Soft well，Intuitive Technology以及Opto22 早在1995年开发了原始的OPC标准，微软同时作为技术顾问给予了支持。

OPC基金会（OPC Foundation，OPC-F），是在1996年9月24日在美国的达拉斯举 行了第一次理事会，并在同年10月7日在美国的芝加哥举行了第一次全体大会上宣告正 式成立的。之后为了普及和进一步改进于1996年8月完成的OPC数据访问标准版本1.0， 开始了全球范围的活动。现在的OPC基金会的理事会是由Fisher-Rosement，Honeywell， Intellution，Rockwell Software，National Instrument以及欧洲代表的Siements和远东代表 的东芝所组成。

在日本为响应以美国为中心的国际标准活动，由１１家公司作为发起人，于1996

年6月开始基金会成立的准备活动，并于1996年10月17日正式成立了日本OPC协会

（OPC-J）。几乎与其同时欧洲OPC协会（OPC-E）也相继成立。在中国也于由５家 公司作为发起人于2001年12月正式成立了中国OPC促进会（OPC-C）。

OPC基金会从成立开始会员逐年增加，到目前为止在全球范围内已有近３００家 公司加入了这个国际标准组织。同时由控制设备厂商和控制软件供应商提供的OPC产 品也日益增加，目前已有６００种以上的OPC服务器产品和OPC应用程序产品出现在 由OPC基金会发行的OPC产品目录上。

**1.1.4 OPC**现状和发展

现存的和正在开发的OPC的标准如表1-1所示。

表 *1-1 OPC*标准

标准 版本 内容

|  |  |  |
| --- | --- | --- |
| Data Access | 1.0A，2.0  3.0正在开发 | 数据访问的标准 |
| Alarm and Events | 1.0  2.0正在开发 | 警报和事件的标准 |
| Historical Data Access | 1.0 | 历史数据访问的标准 |
| Batch | 1.0 | 批处理的标准 |
| Security | 1.0 | 安全性的标准 |
| Compliance | 1.0 | 数据访问标准的测试工具 |
| OPC XML | 正在开发 | 过程数据的XML标准 |
| OPC Data eXchage | 正在开发 | 服务器间数据交换的标准 |

本书是依据OPC数据访问标准（版本2.0），利用微软Visual Basic（以下简称为VB） 或者应用程序的Visual Basic（以下简称为VBA）开发客户应用程序的入门书。本书以 后所提到的“OPC标准”和“OPC服务器”，其实是指OPC数据访问标准（版本2.0） 和OPC数据访问服务器（版本2.0），而“OPC应用程序”是指OPC数据访问客户应用 程序。

**1.2** 什么是**OPC**

**1.2.1** 基于**COM**技术的**OPC**

微软公司为了提供商业应用程序和特定用途的软件包间的相互连接性，开发了所 谓的组件对象模型（Component Object Model, COM）技术。COM是一种软件组件间相 互数据交换的有效方法。COM技术具有以下特长：

 所谓COM并不是一种计算机语言，而是于运行的机器（只要互相连接），机器的 操作系统（只要支持COM），以及软件开发的语言无关，任意的两个软件组件之 间都可以相互通信的二进制和网络的标准。

 COM服务器是根据COM客户的要求提供COM的服务的执行可能的程序，可以作 为Win32上执行可能的文件发布。

 COM客户程序和COM服务器可以用完全不同的语言开发。这样使利用C++，Visual Basic ，以及Excel中作为宏使用的应用程序的Visual Basic 等不同语言所开发的程序 可以相互连接。

 COM组件可以以二进制的形式发布给用户。

 与过去DLL的版本管理非常困难的问题相比，COM技术可以提供不同版本的COM

服务器和COM客户程序之间的最大的兼容性。

 作为COM技术扩展的分布式COM（Distributed Component Object Model, 分布式 COM）技术，更可以使COM组件分布在不同的计算机上，并通过网络互相连接并 互相交换数据。所以对于COM客户程序来说，同样像连接本地计算机上的COM 服务器一样，去连接远程计算机上的COM服务器，当然通信的速度不太一样，但 是重要的是不必对服务器程序进行修正就可以在网络上自由构成（图1-4）。

本地计算机 远程计算机

分布式

COM

客户程序

远程

COM服务器

COM

客户程序

COM接口

远程

COM服务器

COM接口

本地

COM服务器

本地

COM服务器

图 *1-4*利用*COM*和分布式*COM*达成的组件间的互相连接

COM技术的出现使简单地实现控制设备和控制管理系统之间的数据交换提供了 技术基础。但是如果不提供一个工业标准化的COM接口，各个控制设备厂家开发的 COM组件之间的相互连接仍然是不可能的。这样的工业标准的提供，乃是OPC的目的 所在。总而言之，OPC是作为工业标准定义的特殊的COM接口。

OPC是以提供移植容易并具有可以满足大多数设备厂家要求的灵活性和高水平的 机能性为目标而开发的，对于制造厂商和用户来说，分别可以从OPC得到以下的实惠：  设备开发者：可以使设备驱动器开发的单一化成为可能。

 应用程序软件开发者：可以使用通用的开发工具。不必开发特别的接口，使得设 备接口的开发更为简单易行。

 用户：可以选用各种各样的商业软件包，使得系统构成的成本大为降低。同时可 以更加容易地实现由不同供应厂商提供的设备所混合构成的工业控制系统。 随着基于OPC标准的控制组件的推广和普及，不仅使控制系统的增设和组件的置

换就更加简单，而且使过程数据的访问也变得容易。比如，过程控制程序可以直接和 数据分析软件包或电子表格应用程序连接，从而达成高度的工厂控制系统的信息化。 你可能经常听说过COM的编程非常困难，需要高度的计算机知识。确实对于OPC 服务器开发者来说，对 COM的理解和掌握是必不可少的。但是本书的目的是讲述怎样 利用VB或者VBA开发OPC应用程序，所以我们并不要求读者必须掌握COM的知识。 一般来说，只要懂得Visual Basic 的基本使用方法，进行OPC应用程序的编程则是完全

可能的。

**1.2.2 OPC**和**DDE**的比较

在OPC技术出现以前，DDE（Dynamic Data Exchange）技术曾经对过程控制作 出巨大贡献。但是DDE是基于Windows的信息（Message ）传递而建立的技术，所以 DDE技术存在以下问题：

 数据的传送速度较慢

 没有安全性管理机制

 开发困难

 功能缺乏柔软性

 可靠性也难以令人满意 所以基于先进的COM技术的OPC技术将逐渐取代现在在过程控制中广泛使用的

DDE的位置乃是顺理成章的事情。随着OPC技术的导入，和过去的DDE技术相比，在 以下方面显示出它的优越性：

 高速的数据传送性能

 基于分布式COM的安全性管理机制

 开发成本的降低

 实现具有高度柔软性功能的系统

 实现具有高可靠性的系统

图 1-5是分别利用OPC和DDE进行数据传送性能的实验结果的例子。从这里也可以看 出OPC技术在传送速度上的优越性。

OPC DDE

450000

400000

350000

300000

传送项数／秒

250000

200000

150000

100000

50000

0

-1 1 3 5 7 9 11 13 15 远程客户数

图 *1-5* 利用*OPC*和*DDE*的数据传送性能的实验结果

**1.2.3 OPC**适用于哪些地方

OPC是为了连接数据提供源（OPC服务器）和数据的使用者（OPC应用程序）之 间的软件接口标准。

数据提供源可以是PLC，DCS，条形码读取器等控制设备。随控制系统构成的不

同，作为数据提供源的OPC服务器即可以是和OPC应用程序在同一台计算机上运行的 本地OPC服务器，也可以是在另外的计算机上运行的远程OPC服务器。

如图 1-6所示，OPC接口既可以适用于通过网络把最下层的控制设备的原始数据

提供给作为数据的使用者（OPC应用程序）的HMI／SCADA，批处理等自动化程序， 以至更上层的历史数据库等应用程序，也可以适用于应用程序和物理设备的直接连接。 所以OPC接口是适用系统的很多场合的具有高度柔软性的接口标准。

HMI/SCADA OPC 应用程序

历史 数据库

批处理

OPC应用程序

OPC

服务器

OPC

服务器

OPC

服务器

OPC

服务器

设备1

设备2

OPC 服务器

Windows

CE PLC

设备3

设备4 设备5

图 *1-6* 在控制系统中*OPC*所占的位置

**1.3 OPC**的概要

**1.3.1 OPC**服务器和包装**DLL**

利用VB开发的OPC应用程序的系统主要由下列软件组件构成：

1) OPC服务器：OPC服务器一般是由DCS或者I/O驱动器等硬件供应商，或者由独立 的软件供应商提供，即可以在与应用程序计算机相同的本地计算机上运行，也可 以在与应用程序计算机不同的远程计算机上运行。OPC服务器的主要目的是提供 过程数据。

2) OPC代理－占位DLL：因为VB或者VBA的OPC应用程序是运行在与OPC服务器不 同的计算机进程空间，所以不能直接调用OPC服务器的接口进行数据交换，需要 通过代理－占位DLL并利用操作系统提供的通信功能进行数据交换。按照COM的 术语，代理是起着代表别的组件作用的组件的意思。首先在应用程序侧，代理把 向OPC服务器接口传递的数据进行格式变换。然后在OPC服务器侧，占位把OPC 客户程序送过来的数据的格式变换解除，同时也对返回到客户程序的数据进行格 式变换。实际的OPC代理－占位DLL是同一个DLL。随着被设置的计算机的不同， 起着代理的作用或者起着占位的作用。

3) OPC自动化包装DLL： 一般来说，为了达到数据传送的最高的性能，OPC服务

器是用C++开发的，并只提供定制接口。与此相反，用VB或VBA（应用程序的Visual Basic ）等语言开发的应用程序却要求OPC自动化接口。为了让VB或者VBA的客 户应用程序可以使用OPC自动化接口，使用OPC自动化包装将OPC自动化接口变 换成OPC定制接口，从而可以对OPC服务器进行访问。

4) OPC应用程序：对由OPC服务器提供的数据源进行访问，实现用户的特定目的而

开发的应用程序。

利用VB的OPC服务器的配置形态，即可以是和OPC应用程序在同一台计算机内而 在别的进程中运行的本地OPC服务器，也可以是通过网络在另外的计算机上运行的远 程OPC服务器。图 1-7和图 1-8分别表示本地和远程OPC服务器的配置形态。

本地计算机

OPC应用程序

包装DLL

包装DLL

OPC应用程序

OPC自动化接口

OPC定制接口

代理-占位DLL

代理-占位DLL

计算机过程边界

代理-占位DLL

OPC定制接口

OPC服务器

代理-占位DLL

OPC服务器

图 *1-7* 本地*OPC*服务器

本地计算机

OPC应用程序

包装DLL

包装DLL

OPC应用程序

OPC自动化接口

OPC定制接口

代理-占位DLL

计算机网络边界

远程计算机

代理-占位DLL

代理-占位DLL

代理-占位DLL

OPC定制接口

OPC服务器

OPC服务器

图 *1-8* 远程*OPC*服务器

OPC服务器，OPC代理－占位DLL以及OPC包装DLL应该由OPC服务器供应商（控 制设备制造商或者独立软件供应商）提供的，并不在本书的说明范围之内。本书的目 的主要是解说怎样利用VB或者VBA开发OPC的应用程序。

**1.3.2 OPC**的主要功能

OPC的数据访问方法主要有同步访问和异步访问两种。

对于如图 1-9所示的同步访问，OPC服务器把按照OPC应用程序的要求得到的数 据访问结果作为方法的参数返回给OPC应用程序，OPC应用程序在结果被返回为止一 直必须处于等待状态。

方法调用开始 方法调用结束

OPC应用程序 时间

等待状态

调用同步访问方法

OPC服务器 时间

图 *1-9* 同步数据访问处理

与此相对，如图 1-10所示的异步访问，OPC服务器接到OPC应用程序的要求后， 几乎立即将方法返回。OPC应用程序随后可以进行其他处理。当OPC服务器完成数据 访问时，触发OPC应用程序的异步访问完成事件，将数据访问结果传送给OPC应用程 序。OPC应用程序在VB的事件处理程序中接受从OPC服务器传送来的数据。

方法调用开始 方法调用结束

事件发生 事件处理结束

OPC应用程序 时间

等待 其他处理 通信

调用 异步 访问方法

异步访问完 成事件

OPC服务器 时间

图 *1-10* 异步数据访问处理

除了上述的同步和异步数据访问以外，还有如图1-11所示的并不需要OPC应用程 序向OPC服务器要求，就可以自动接到从OPC服务器送来的变化通知的订阅方式数据 采集（Subscription）。服务器按一定的更新周期（UpdateRate ）更新OPC服务器的数 据缓冲器的数值时，如果发现数值有变化时，就会以数据变化事件（DataChange）通 知OPC应用程序。如果OPC服务器支持不敏感带（DeadBand），而且OPC标签的数据 类型是模拟量的情况，只有现在值与前次值的差的绝对值超过一定限度时，才更新缓 冲器数据并通知OPC应用程序。由此可以无视模拟值的微小变化，从而减轻OPC服务 器和OPC应用程序的负荷。

事件发生 事件处理结束

事件发生 事件处理结束

OPC应用程序

其他处理

通知 其他处理 通知

时间

其他处理

数据变化事件

(DataChange)

数据变化事件

(DataChange)

OPC服务器 时间

采样周期

图 *1-11* 订阅方式数据采集

上述的OPC数据访问的功能可总结成表1-2。这些功能并非必须全部实现，其中有 一部分功能是选用的。这些选用功能是否被支持将随供应厂商的具体的服务器类型而 定。

表 *1-2 OPC*数据访问的功能

|  |  |  |
| --- | --- | --- |
| 功能 | 方式 | 说明 |
| 过程数据读取  ※ 这是独立的互不影响 的四种读取方式。 | 同步读取 | 读取指定OPC标签对应的过程数据。  应用程序一直等待到读取完成为止。 |
| 异步读取 | 读取指定OPC标签对应的过程数据。  应用程序发出读取要求后立即返回， 读取完成时发生读取完成事件，OPC 应用程序被调出。 |
| 刷新 | 读取所有活动的OPC标签对应的过程  数据。 应用程序发出更新要求后立即返回， 更新完成时发生数据变化事件，OPC 应用程序被调出。 |
| 订阅方式数据采集  （Subscription） | 服务器用一定的周期检查过程数据，  如发现数据变化超过一定的幅度时， 则更新数据缓冲器，并自动通知OPC 应用程序。 |
| 过程数据写入 | 同步写入 | 写入指定OPC标签对应的过程数据。  应用程序一直等待到写入完成为止。 |

异步写入 写入指定OPC标签对应的过程数据。 应用程序发出写入要求后立即返回， 写入完成时发生写入完成事件，OPC 应用程序被调出。

同步和异步数据访问的特征可总结如表1-3。OPC应用程序的开发者可按照应用程 序的用途和目的选择合适的数据访问方法。

表 *1-3 OPC*同步和异步数据访问的特征

|  |  |  |
| --- | --- | --- |
| 特征 | 同步访问 | 异步访问 |
| 访问性能 | 因为在访问完成之前应用程序  必须一直在等待，尤其大量数 据的访问或直接向设备的访问 对访问性能的影响很大。 | 因为在访问完成之前应用程  序不必等待，可以并行处理， 对访问性能的影响不大。 |
| 程序开发 | 处理程序比较简单，开发容易。 | 因为发出要求和访问完成事  件处理是分别进行的，所以必 须有事务（Transaction）识别 功能，开发比较难。 |
| 远程连接的分布式  COM设置 | 只要分布式COM启动权限和访  问权限就可以运行，设置比较 简单。 | 除了分布式COM启动权限和  访问权限以外，还必须设置身 份标志，设置比较复杂。 |

**1.4 OPC**的对象

OPC应用程序应该首先生成OPC服务器支持的OPC对象，然后就可以使用OPC对 象支持的属性和方法，对其进行简单的操作。这种结构使得应用程序可以象使用自己 支持的数据和功能一样，去使用服务器对象支持的数据和功能。例如，如图 1-13所示， OPC应用程序可以取得OPC服务器支持属性的ServerState即服务器执行状态，还可以 调用OPC服务器支持的方法Connect()以和服务器连接。

OPC 应用程序 OPC 服务器

ServerState

……

Connect()

……

数据(属性)

公开 ServerState

功能(方法)

公开 Connect()

图 *1-13 OPC*应用程序和*OPC*服务器

注意一个OPC应用程序可以与多个的服务器同时连接，同时一个OPC服务器也可以同 时被多个的OPC应用程序连接。

**1.4.1 OPC**对象的分层结构

OPC数据访问提供从数据源读取和写入特定数据的手段。OPC数据访问对象是由 象图 1-14所示的分层结构构成。即一个OPC服务器对象（OPCServer）具有一个作为 子对象的OPC组集合对象（OPCGroups）。在这个OPC组集合对象里可以添加多个的 OPC组对象（OPCGroup）。各个OPC组对象具有一个作为子对象的OPC标签集合对象

（OPCItems ）。在这个OPC标签集合对象里可以添加多个的OPC标签对象（OPCItem）。 此外，作为选用功能，OPC服务器对象还可以包含一个OPC浏览器对象（OPCBrowser）。

服务器(OPCServer)

服务器(OPCServer)

组集合(OPCGroups)

浏览器(OPCBrowser )

组( OPCGroup)

组集合(OPCGroups)

? ? ? ? ? ? （**OPCGroup**）

项集合(OPCItems)

浏览器(OPCBrowser)

项集合(OPCItems)

? ? ? ? （**OPCItem**）

? ? ? ? （**OPCItem**）

? ? ? ? ? ? （**OPCGroup**）

? ? ? ? ? ? （**OPCGroup**）

? ? ? ? ? ? （**OPCGroup**）

组( OPCGroup)

项(OPCItem)

? ? ? ? （**OPCItem**）

图 *1-14 OPC*数据访问对象的分层结构

? ? ? ? （**OPCItem**）

项(OPCItem)

OPC对象中的最上层的对象是OPC服务器。一个OPC服务器里可以设置一个以上 的OPC组。OPC服务器经常对应于某种特定的控制设备。例如，某种DCS控制系统， 或者某种PLC控制装置。

OPC组是可以进行某种目的数据访问的多个的OPC标签的集合，例如某监视画面 里所有需要更新的位号变量。正因为有了OPC组，OPC应用程序就可以以同时需要的 数据为一批的进行数据访问，也可以以OPC组为单位启动或停止数据访问。此外OPC 组还提供组内任何OPC标签的数值变化时向OPC应用程序通知的数据变化事件（表

1-4）。

表 *1-4 OPC*数据访问对象模型

|  |  |  |
| --- | --- | --- |
| 名称 | 对象名 | 说明 |
| OPC服务器 | OPCServer | OPC服务器对象在使用其他OPC对象前必须生成。OPC服务  器自动含有一个OPC组集合对象，并可在其基础上生成一个  OPC浏览器对象。 |
| OPC组集合 | OPCGroups | OPC服务器中添加的所有的OPC组的集合。 |
| OPC组 | OPCGroup | OPC组对象是用于组的状态管理以及利用项集合为单位的  数据访问。 |
| OPC标签集  合 | OPCItems | 在对应OPC组中添加的所有的OPC标签的集合。 |
| OPC标签 | OPCItem | 含有OPC标签的定义，现在值，状态，以及最后更新时间等  信息的对象。 |
| OPC浏览器 | OPCBrowse r | 用于浏览OPC服务器的名称空间的对象。 |

**1.4.2 OPC**标签

OPC对象里最基本的对象是OPC标签。OPC标签是OPC服务器可认识的数据定义， 通常相当于位号的单一变量（调整点或过程数据），并和数据提供源（控制设备）相 连接。OPC标签具有多个属性，但是其中最重要的属性是OPC标签标识符。项标识符 是在控制系统中可识别OPC标签的字符串，例如，

DCS的例："TIC100.PV" PLC的例："COM1.STATION:42.REG:40001;0,4095,-100,+1234.0"

**1.4.3** 服务器句柄

一旦OPC组或者OPC标签在OPC服务器里添加成功，OPC服务器将赋予被添加的 各个OPC组或者各个OPC标签一个独特的标识符。这个标识符叫做服务器句柄。被赋 予的服务器句柄将返回给OPC应用程序。

OPC应用程序应该将由OPC服务器返回的OPC组或者OPC标签的服务器句柄好好

保管。因为随后对添加的OPC组或者OPC标签进行操作时，只有使用这些服务器句柄 才可以唯一地识别特定的OPC组或者OPC标签。

**1.4.4 OPC**服务器对象

因为OPC服务器对象OPCServer提供连接数据源（OPC定值接口服务器）以及数 据访问（读取· 写入）的方法，所以在建立OPC组和OPC标签以前必须建立OPC服务 器对象，然后使用OPC数据访问自动化接口的“Connect”方法和数据源连接。本节说 明一些经常使用的OPC服务器的属性和方法。本节的示例程序需要表1-5的示例基础。

表 *1-5 OPC*服务器的示例基础

示例 Dim WithEvents AnOPCServer As OPCServer

基础

Set AnOPCServer = New OPCServer

属性：ServerState

|  |  |
| --- | --- |
| 说明 | 只读的属性，返回服务器的运行状态（OPCServerStatus ）。 |
| 文法 | ServerState As Long  下列的OPCServerStatus 值是可能的： OPCRunning：OPC服务器正在正常运转。 OPCFailed：OPC服务器由于异常而停止。  OPCNoconfig ：OPC服务器正在运转，但没有被设置。 OPCSuspended：OPC服务器正处于暂时停止状态。 OPCTest：OPC服务器正在实验模式下运转。 OPCDisconnected：服务器对象没有连接任何实际的OPC服务器。 |
| 示例 | Dim ServerState As Long  ServerState = AnOPCServer.ServerState |

属性：OPCGroups

|  |  |
| --- | --- |
| 说明 | 只读的属性，OPC组的集合。这是OPCServer 的默认属性。 |
| 文法 | OPCGroups As OPCGroups |
| 示例 | Dim MyGroups As OPCGroups  ‘ 使用显然的属性声明  Set MyGroups = AnOPCServer.OPCGroups  ‘ 或使用默认的属性声明  Set Mygroups = AnOPCServer |

方法：Connect

|  |  |
| --- | --- |
| 说明 | 连接OPC数据访问服务器。 |
| 文法 | Connect (ProgID As String, Optional Node As Variant)  ProgID: 程序标识符是可以识别特定OPC服务器的注册字符串。  Node: 选用参数，是利用分布式COM进行远程连接的计算机UNC名称（例如， "Server"）或者DNS名称（例如，w ww.vendor.com或者"180.151.19.75"）。 被省略时，将连接本地OPC服务器。 |
| 示例 | Dim ProgID As String, NodeName As String  ProgID = "VendorX.DataAccessCustomServer" NodeName = "SomeComputerNodeName" AnOPCServer.Connect (ProgID, NodeName) |

方法：Disconnect

|  |  |
| --- | --- |
| 说明 | 断开和OPC服务器的连接。OPC应用程序在断开和OPC服务器的连接前，建  议显式的清除所有添加的OPC组和OPC标签的程序，虽然调用本方法也可以 黯然地清除所有的OPC组，并释放所有的引用。 |
| 文法 | Disconnect () |

示例 AnOPCServer.Disconnect

事件：ServerShutDown

|  |  |
| --- | --- |
| 说明 | 这个事件在服务器即将关机前发生，OPC服务器以此事件通知OPC应用程序  预告即将关机。OPC应用程序应该在接到此事件通知后，立即清除所有的  OPC组并断开与OPC服务器连接。 |
| 文法 | ServerShutDown (Reason As String) ServerReason是选用的参数，是说明服务器关机理由的字符串。 |
| 示例 | Dim WithEvents AnOPCServer As OPCServer  Dim ProgID As String, NodeName As String  ProgID = "VendorX.DataAccessCustomServer" NodeName = "SomeComputerNodeName" AnOPCServer.Connect (ProgID, NodeName)  Private Sub AnOPCServer\_ServerShutDown(ByRef aServerReason As String)  ‘ 在这儿记述服务器关机处理代码  Debug.Print aServerReason AnOPCServer. OPCGroups.RemoveAll AnOPCServer.Disconnect  End Sub |

所有OPC服务器支持的属性,方法和事件的一览如表1-6，表1-7和表1-8所示。详细 资料请参照6.5参考资料的“OPC数据访问自动化接口标准”。

表 *1-6 OPC*服务器的属性

|  |  |  |
| --- | --- | --- |
| 属性名 | 属性 | 说明 |
| StartTime | 只读 | OPC服务器的启动时间（国际标准时间）。 |
| CurrentTime | 只读 | OPC服务器的现在时间（国际标准时间）。 |
| LastUpdateTime | 只读 | 对于本OPC应用程序的最后数据更新时间（国际  标准时间）。 |
| MajorVersion | 只读 | OPC服务器软件的主版本号。 |
| MinorVersion | 只读 | OPC服务器软件的次版本号。 |
| BuildNumber | 只读 | OPC服务器软件的版本生成号。 |
| VendorInfo | 只读 | 开发者提供的有关OPC服务器软件版本信息的  文字串。（建议含有公司名以及所支持的设备类 型） |
| ServerState | 只读 | 返回OPC服务器的运行状态（OPCServerStatus ）。 |
| LocaleID | 读写 | 返回区域标识符（语言标识符）。 |
| BandWidth | 只读 | OPC服务器的特有值。返回OPC服务器使用可能  的不敏感带的百分比（%）。 |
| OPCGroups | 只读 | OPC的组集合。服务器的默认属性。 |
| PublicGroupNames | 只读 | 返回OPC服务器的OPC公用组名称。 |
| ServerName | 只读 | 返回OPC服务器的名称。 |
| ServerNode | 只读 | 返回OPC服务器的计算机名。 |
| ClientName | 读写 | OPC应用程序在OPC服务器中的注册名。用于调  试。 |

表 *1-7 OPC*服务器的方法

|  |  |
| --- | --- |
| 方法名 | 说明 |
| GetOPCServers | 返回注册的OPC服务器的程序标识符（ProgID）。 |
| Connect | 连接OPC数据访问服务器。 |
| Disconnect | 断开和OPC服务器的连接。 |
| CreateBrowser | 生成OPC浏览器对象。 |
| GetErrorString | 得到按区域标识符（LocaleID）指定的错误码文字说明。 |
| QueryAvailableLocaleIDs | 询问在服务器和应用程序之间使用可能的区域标识符  （LocaleID）。 |
| QueryAvailableProperties | 对于指定的项属性，询问可能取得的项属性的标识符和  文字说明。 |
| GetItemProperties | 按照指定的OPC标签和属性标识符，取得项属性的现在  值。 |
| LookUpItemIDs | 按照指定的OPC标签和属性标识符，取得与项属性值对  应的OPC标签标识符。 |

表 *1-8 OPC*服务器的事件

|  |  |
| --- | --- |
| 事件 | 说明 |
| ServerShutDown | 这个事件在服务器即将关机前发生，OPC服务器以此事  件通知OPC应用程序预告即将关机。OPC应用程序应该 在接到此事件通知后，立即清除所有的OPC组并断开与 服务器连接。 |

**1.4.5 OPC**组集合对象

OPC组集合对象OPCGroups是OPC组的集合，这个对象的用途是添加，清除和管 理OPC组。本节说明一些经常使用的OPC组集合的方法。本节的示例程序需要表1-9 的示例基础。

表 *1-9 OPC*组集合的示例基础

示例 Dim WithEvents AnOPCServer As OPCServer

基础 Dim ProgID As String

Dim NodeName As String

Dim MyGroups As OPCGroups

Dim OneGroup As OPCGroup

Set AnOPCServer = New OPCServer

ProgID = "VendorX.DataAccessCustomServer" NodeName = "SomeComputerNodeName" AnOPCServer.Connect(ProgID, NodeName)

Set MyGroups = AnOPCServer.OPCGroups

方法：Add

|  |  |
| --- | --- |
| 说明 | 在OPC组集合中建立新的OPC组。 |
| 文法 | Add(Optional Name As Variant) As OPCGroup  Name是独特的OPC组名。这是选用参数，没有被定义时，由OPC服务器自动 产生组名。 |
| 示例 | Set OneGroup = MyGroups.Add("AnOPCGroupName" ) |

方法：Remove

|  |  |
| --- | --- |
| 说明 | 清除指定的OPC组。 |
| 文法 | Remove(ItemSpecifier As Variant)  ItemSpecifier 是要清除的OPC组的服务器句柄或者OPC组名。 |
| 示例 | Set OneGroup = MyGroups.Add("AnOPCGroupName" )  ‘ 使用OPC组名的处理代码  MyGroups.Remove("AnOPCGroupName" )  ‘ 或者使用组服务器句柄的处理代码  MyGroups.Remove(OneGroup.ServerHandle ) |

方法：RemoveAll

|  |  |
| --- | --- |
| 说明 | 为服务器关机作准备，清除所有的OPC组和OPC标签。 |
| 文法 | RemoveAll() |
| 示例 | Set OneGroup = MyGroups.Add("AnOPCGroupName" ) Set OneGroup = MyGroups.Add("AnOPCGroupName1" ) Set OneGroup = MyGroups.Add("AnOPCGroupName2")  MyGroups.RemoveAll |

所有OPC组集合支持的属性，方法和事件的一览如表1-10，表1-11和表1-12所示。详细 资料请参照6.5参考资料的“OPC数据访问自动化接口标准”。

表 *1-10 OPC*组集合的属性

|  |  |  |
| --- | --- | --- |
| 属性名 | 属性 | 说明 |
| Parent | 只读 | 返回所属OPC服务器对象。 |
| DefaultGroupIsActive | 读写 | 新添加的OPC组的活动状态的默认值。初期值是  活动状态。 |
| DefaultGroupUpdateRate | 读写 | 新添加的OPC组的更新周期的默认值（毫秒）。  初期值是1000毫秒。 |
| DefaultGroupDeadBand | 读写 | 新添加的OPC组的不敏感带的默认值（%）。初  期值是0%。 |
| DefaultGroupLocaleID | 读写 | 新添加的OPC组区域标识符的默认值。 |
| DefaultGroupTimeBias | 读写 | 新添加的OPC组的时间偏差的默认值（分钟）。  初期值是0分。 |
| Count | 只读 | 集合对象的固有属性。包含的组数。 |

表 *1-11 OPC*组集合的方法

|  |  |
| --- | --- |
| 方法名 | 说明 |
| Item | OPC组集合的默认方法。返回由集合索引（ItemSpecifier ）  指定的OPC组对象。 |
| Add | 在OPC组集合中添加新的OPC组。 |
| GetOPCGroup | 返回指定的OPC组对象。 |
| Remove | 清除指定的OPC组。 |
| RemoveAll | 为服务器关机作准备，清除所有的OPC组和OPC标签。 |
| ConnectPublicGroup | 连接OPC公用组。 |
| RemovePublicGroup | 清除OPC公用组。 |

表 *1-12 OPC*组集合的事件

|  |  |
| --- | --- |
| 事件 | 说明 |
| AllGroupsDataChange | 由多个OPC组的数据变化而引发的事件 |

**1.4.6 OPC**组对象

OPC组对象OPCGroup提供满足OPC应用程序要求的数据访问手段。本节说明一些 经常使用的OPC服务器的属性，方法和事件。本节的示例程序需要表1-13的示例基础。

表 *1-13 OPC*组的示例基础

示例 Dim WithEvents AnOPCServer As OPCServer

基础 Dim ProgID As String

Dim NodeName As String

Dim MyGroups As OPCGroups

Dim WithEvents OneGroup As OPCGroup Dim AnOPCItemCollection As OPCItems Dim AnOPCItem As OPCItem

Dim ClientHandles(10) As Long

Dim AnOPCItemIDs(10) As String

Dim AnOPCItemServerHandles() As Long Dim AnOPCItemServerErrors() As Long Dim AddItemCount As Long

Dim I As Long

Set AnOPCServer = New OPCServer

ProgID = "VendorX.DataAccessCustomServer" NodeName = "SomeComputerNodeName" AnOPCServer.Connect(ProgID, NodeName)

Set MyGroups = AnOPCServer.OPCGroups

Set OneGroup = MyGroups.Add("AnOPCGroupName" )

Set AnOPCItemCollection = OneGroup.OPCItems

AddItemCount = 10

For I = 1 To AddItemCount

ClientHandles(I) = I + 1

AnOPCItemID(I) = "Item\_" & I

Next

‘ 添加OPC标签

AnOPCItemCollection.AddItems AddItemCount, AnOPCItemIDs, \_ ClientHandles, AnOPCItemServerHandles, AnOPCItemServerErrors

属性：IsActive

|  |  |
| --- | --- |
| 说明 | 可设置的属性，用以控制OPC组的活动状态。只有处于活动状态的OPC组才  进行定期的数据更新。非活动状态的OPC组，除了在接到显然的数据读写要 求外，并不收集任何数据。 |
| 文法 | IsActive As Boolean |
| 示例 | Dim CurrentValue As Boolean  ‘ 设置OPC组为活动状态 |

OneGroup.IsActive = True

‘ 读取OPC组的活动状态

CurrentValue = OneGroup.IsActive

属性：IsSubscribed

|  |  |
| --- | --- |
| 说明 | 可设置的属性，用以控制OPC组的订阅状态。进行订阅的OPC组可以自动收  到从服务器送来的数据变化通知。 |
| 文法 | IsSubscribed As Boolean |
| 示例 | Dim CurrentValue As Boolean  ‘ 设置OPC组为订阅状态  OneGroup.IsSubscribed = True  ‘ 取得OPC组的订阅状态  CurrentValue = OneGroup.IsSubscribed |

属性：ServerHandle

|  |  |
| --- | --- |
| 说明 | 只读的属性，服务器句柄是由OPC服务器指定的，用于识别指定的OPC组的  一个独特的长整型数。OPC应用程序可以利用这个服务器句柄，向OPC服务 器要求对指定的OPC组进行操作，例如清除指定的OPC组。 |
| 文法 | ServerHandle As Long |
| 示例 | Dim CurrentValue As Long  ‘ 读取OPC组的服务器句柄  CurrentValue = OneGroup.ServerHandle |

属性：UpdateRate

|  |  |
| --- | --- |
| 说明 | 可设置的属性，以毫秒为单位的数据更新周期。 |
| 文法 | UpdateRate As Long |
| 示例 | Dim CurrentValue As Long  ‘ 设置OPC组的更新周期为５秒  OneGroup.UpdateRate = 5000  ‘ 取得OPC组的更新周期  CurrentValue = OneGroup.UpdateRate |

属性：OPCItems

|  |  |
| --- | --- |
| 说明 | 只读的OPC组的默认属性，OPC标签集合对象。 |
| 文法 | OPCItems As OPCItems |
| 示例 | ‘ 取得OPC标签集合对象  Set AnOPCItemCollection = OneGroup.OPCItems |

方法：SyncRead

|  |  |
| --- | --- |
| 说明 | 同步读取OPC组内单个或者多个OPC标签的数据值，质量标志和采样时间。 |
| 文法 | SyncRead(Source As Integer, NumItems As Long, \_ ServerHandles() As Long, ByRef Values() As Variant, \_  ByRef Errors() As Long, Optional ByRef Qualities As Variant, \_ Optional ByRef, TimeStamps As Variant) |

|  |  |
| --- | --- |
|  | Source 数据源。可以指定为OPCCache（缓冲器）或者OPCDevice（设备）。  NumItems 要读取的OPC标签的数目。  ServerHandles 要读取的OPC标签的服务器句柄的数组。  Values 返回的读取的数值的数组。  Errors 返回的与读取项对应的错误码的数组。 Qualities 选用参数，读取数值的质量标志的数组。 TimeStamps 选用参数，读取数据的采样时间的数组。 |
| 示例 | Dim NumItems As Long  Dim ServerHandles(10) As Long  Dim Values() As Variant Dim Errors() As Long Dim Qualities As Variant  Dim TimeStamps As Variant  Dim I As Long  NumItems = 10  ‘ 设置要读取的OPC标签的服务器句柄  For I = 1 to NumItems  ServerHandles(I) = AnOPCItemServerHandles(I) Next  ‘ 同步读取  OneGroup.SyncRead OPCDevice, NumItems, \_ ServerHandles, Values, Errors, Qualities, TimeStamps  ‘ 表示读取OPC标签的数值  For I = 1 to NumItems  Debug.Print I;Values(I) Next |

方法：SyncWrite

|  |  |
| --- | --- |
| 说明 | 同步写入OPC组内单个或者多个OPC标签的数据值。因为数据被直接同步地  写入到设备中，所以只有等数据被设备接受或拒绝后，这个方法的调用才会 结束。 |
| 文法 | SyncWrite(NumItems As Long, ServerHandles() As Long, \_ Values() As Variant, ByRef Errors() As Long)  NumItems 要写入的OPC标签的数目。  ServerHandles 要写入的OPC标签的服务器句柄的数组。  Values 要写入的数值的数组。  Errors 返回的与写入项对应的错误码的数组。 |
| 示例 | Dim NumItems As Long  Dim ServerHandles() As Long  Dim Values() As Variant Dim Errors() As Long Dim I As Long  NumItems = 10  ‘ 设置要写入的OPC标签的服务器句柄  For I = 1 to NumItems  ServerHandles(I) = AnOPCItemServerHandles(I) |

Values(I) = I \* 2

Next

‘ 同步写入

OneGroup.SyncWrite NumItems, ServerHandles, Values, Errors

‘ 表示写入OPC标签的错误码

For I = 1 to NumItems

Debug.Print I; Errors(I) Next

方法：AsyncRead

|  |  |
| --- | --- |
| 说明 | 异步读取OPC组内单个或者多个OPC标签的数据值，质量标志和采样时间。  利用异步数据访问时，必须将OPC组声明为可响应事件的对象变量（Dim WithEvents xxx As OPCGroup）。读取结果是由AsyncReadComplete事件返回。 请注意因为本方法的数据是直接从设备中读取的，所以并不受到OPC组的活 动状态的影响。 |
| 文法 | AsyncRead( NumItems As Long, ServerHandles() As Long, \_  ByRef Errors() As Long, TransactionID As Long, ByRef CancelID As Long)  NumItems 读取的OPC标签的数目。  ServerHandles 要读取的OPC标签的服务器句柄的数组。  Errors 返回的与读取项对应的错误码的数组。  TransactionID 由OPC应用程序发行的事务标识符。当数据访问完成后事件发 生时，OPC应用程序可以利用这个事务标识符识别所完成的异步数据访问。 CancelID 由服务器发行的取消标识符。OPC应用程序使用这个标识符，可以 取消正在进行中的异步数据访问。 |
| 示例 | Dim NumItems As Long  Dim ServerHandles(10) As Long  Dim Errors() As Long  Dim ClientTransactionID As Long Dim ServerTransactionID As Long Dim I As Long  NumItems = 10  ClientTransactionID = 1975  ‘ 设置要读取的OPC标签的服务器句柄  For I = 1 to NumItems  ServerHandles(I) = AnOPCItemServerHandles(I) Next  ‘ 异步读取  OneGroup.AsyncRead NumItems, ServerHandles, Errors, \_ ClientTransactionID, ServerTransactionID |

方法：AsyncWrite

|  |  |
| --- | --- |
| 说明 | 异步写入OPC组内单个或者多个OPC标签的数据值。利用异步数据访问时，  必须将OPC组声明为可响应事件的对象变量（Dim WithEvents xxx As  OPCGroup）。写入结果是由AsyncWriteComplete事件返回。 |
| 文法 | AsyncWrite(NumItems As Long, ServerHandles() As Long, \_  Values() As Variant, ByRef Errors() As Long, TransactionID As Long, \_ ByRef CancelID As Long) |

|  |  |
| --- | --- |
|  | NumItems 要写入的OPC标签的数目。  ServerHandles 要写入的OPC标签的服务器句柄的数组。  Values 要写入的数值的数组。  Errors 返回的与写入项对应的错误码的数组。  TransactionID 由OPC应用程序发行的事务标识符。当数据访问完成后事件发 生时，OPC应用程序可以利用这个事务标识符识别所完成的异步数据访问。 CancelID 由服务器发行的取消标识符。OPC应用程序使用这个标识符，可以 取消正在进行中的异步数据访问。 |
| 示例 | Dim NumItems As Long  Dim ServerHandles(10) As Long  Dim Values() As Variant  Dim Errors() As Long  Dim ClientTransactionID As Long Dim ServerTransactionID As Long Dim I As Long  NumItems = 10  ClientTransactionID = 1957  ‘ 设置要写入的OPC标签的服务器句柄  For I = 1 to NumItems  ServerHandles(I) = AnOPCItemServerHandles(I) Values(I) = I \* 2  Next  ‘ 异步写入  OneGroup.AsyncWrite NumItems, ServerHandles, Values, Errors, \_ ClientTransactionID , ServerTransactionID |

事件：DataChange

|  |  |
| --- | --- |
| 说明 | 在OPC组内任何OPC标签的数据值或者质量标志变化时触发的事件。但不会  在下次OPC组的更新周期（UpDateRate）以前发生。注意订阅方式数据采集  （Subscription）和异步的数据刷新（AsyncRefresh）都可以触发这个事件， 但是不同的是由订阅方式数据采集触发的事件返回的事务标识符为零  （TransactionID = 0），而由异步数据刷新触发的事件返回的事务标识符非 零（TransactionID ? 0）。 |
| 文法 | DataChange (TransactionID As Long, NumItems As Long, \_ ClientHandles() As Long, Values() As Variant, Qualities() As Long, \_ TimeStamps() As Date)  TransactionID 由OPC应用程序发行的事务标识符。事务标识符为零的是订阅 方式数据采集（Subscription）的返回结果，而事务标识符非零的是异步的数 据刷新（AsyncRefresh）的返回结果。  NumItems 读取的OPC标签的数目。  ClientHandles 读取的OPC标签的客户句柄的数组。  Values 返回的读取的数值的数组。 Qualities 读取的质量标志的数组。 TimeStamps 读取的采样时间的数组。 |
| 示例 | Private Sub AnOPCGroup\_DataChange (TransactionID As Long, \_ NumItems As Long, ClientHandles() As Long, Values() As Variant, \_ Qualities() As Long, TimeStamps() As Date)  Dim I As Long |

‘ 表示读取OPC标签的客户句柄和数值

For I = 1 to NumItems

Debug.Print I; ClientHandles(I); Values(I) Next

End Sub

事件：AsyncReadComplete

|  |  |
| --- | --- |
| 说明 | 在异步读取（AsyncRead）完成时发生的事件。 |
| 文法 | AsyncReadComplete (TransactionID As Long, NumItems As Long, \_ ClientHandles() As Long, Values() As Variant, Qualities() As Long, \_ TimeStamps() As Date, Errors() As Long)  TransactionID 由OPC应用程序发行的事务标识符。  NumItems 读取的OPC标签的数目。  ClientHandles 读取的OPC标签的客户句柄的数组。  Values 返回的读取的数值的数组。 Qualities 读取的质量标志的数组。 TimeStamps 读取的采样时间的数组。 |
| 示例 | Private Sub AnOPCGroup\_AsyncReadComplete (TransactionID As Long, \_ NumItems As Long, ClientHandles() As Long, ItemValues() As Variant, \_ Qualities() As Long, TimeStamps() As Date)  Dim I As Long  ‘ 表示读取OPC标签的客户句柄和数值  For I = 1 to NumItems  Debug.Print I; ClientHandles(I); Values(I) Next  End Sub |

事件：AsyncWriteComplete

|  |  |
| --- | --- |
| 说明 | 在异步写入（AsyncWrite）完成时发生的事件。 |
| 文法 | AsyncWriteComplete (TransactionID As Long, NumItems As Long, \_ ClientHandles() As Long, Errors() As Long)  TransactionID OPC应用程序发行的事务标识符。  NumItems 写入的OPC标签数。  ClientHandles 写入的OPC标签的客户句柄の的数组。  Errors 返回的与写入项对应的错误码的数组。 |
| 示例 | Private Sub AnOPCGroup\_AsyncWriteComplete (TransactionID As Long, NumItems As Long, ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long, TimeStamps() As Date)  Dim I As Long  ‘ 表示写入错误码  For I = 1 to NumItems  Debug.Print I; Errors(I) Next  End Sub |

所有OPC组支持的属性，方法和事件的一览如表1-14，表1-15和表1-16所示。详细资料 请参照6.5参考资料的“OPC数据访问自动化接口标准”。

表 *1-14 OPC*组的属性

|  |  |  |
| --- | --- | --- |
| 属性名 | 属性 | 说明 |
| Parent | 只读 | 返回所属OPC服务器对象。 |
| Name | 读写 | OPC组的名称。 |
| IsPublic | 只读 | OPC组是否是公用组的真伪值。 |
| IsActive | 读写 | 用以控制OPC组的活动状态。只有活动状态的  OPC组才进行定期的数据更新。 |
| IsSubscribed | 读写 | 用以控制OPC组的订阅状态。 |
| ClientHandle | 读写 | 客户句柄是由应用程序指定的用于识别某个  OPC组的长整型数。当进行数据访问或询问OPC 组状态时，服务器将这个数值和结果一起返回给 OPC应用程序。 |
| ServerHandle | 只读 | 服务器句柄是由OPC服务器指定的用于识别某  个OPC组的一个独特的长整型数。 |
| LocaleID | 读写 | 区域标识符。 |
| TimeBias | 读写 | 数据采样时间的时间偏差值，用于调整设备时间  和OPC服务器时间之间的偏差。 |
| DeadBand | 读写 | 不敏感带（全量程的百分比；合法值从0到100）。  只有数据变化超过此不敏感带时，服务器才触发 数据变化事件发生。 |
| UpdateRate | 读写 | 数据更新周期（毫秒）。 |
| OPCItems | 只读 | OPC组的默认属性，OPC标签集合对象。 |

表 *1-15 OPC*组的方法

|  |  |
| --- | --- |
| 方法名 | 说明 |
| SyncRead | 同步读取OPC组内单个或者多个OPC标签的数据值，质  量标志和采样时间。 |
| SyncWrite | 同步写入OPC组内单个或者多个OPC标签的数据值 |
| AsyncRead | 异步读取OPC组内单个或者多个OPC标签的数据值，质  量标志和采样时间。 |
| AsyncWrite | 异步写入OPC组内单个或者多个OPC标签的数据值。 |
| AsyncRefresh | 触发数据变化事件发生，刷新OPC组内所有活动的OPC  标签的数据。结果由数据变化（ DataChange）事件返回。 |
| AsyncCancel | 取消尚未完成的异步数据访问事务。处理结果由异步取  消完成（AsyncCancelComplete）事件返回。 |

表 *1-16 OPC*组的事件

|  |  |
| --- | --- |
| 事件名 | 说明 |
| DataChange | 在OPC组内任何OPC标签的数据值或者质量标志变化  时触发的事件。 |
| AsyncReadComplete | 在异步读取（AsyncRead）完成时发生的事件。 |
| AsyncWriteComplete | 在异步写入（AsyncWrite）完成时发生的事件。 |
| AsyncCancelComplete | 在取消异步访问（AsyncCancel）完成时发生的事件。 |

**1.4.7 OPC**标签集合对象

OPC标签集合对象OPCItems 具有OPC标签的默认属性，当添加新的OPC标签时， 下述的DefaltXXX属性将是新添加的OPC标签的默认属性值。本节说明一些经常使用 的OPC标签集合的属性和方法。本节的示例程序需要表1-17的示例基础。

表 *1-17 OPC*标签集合的示例基础

示例 Dim WithEvents AnOPCServer As OPCServer

基础 Dim ProgID As String

Dim NodeName As String

Dim MyGroups As OPCGroups

Dim WithEvents OneGroup As OPCGroup Dim AnOPCItemCollection As OPCItems Dim AnOPCItem As OPCItem

Dim AnOPCItemServerHandles() As Long

Dim AnOPCItemServerErrors() As Long

Set AnOPCServer = New OPCServer

ProgID = "VendorX.DataAccessCustomServer" NodeName = "SomeComputerNodeName" AnOPCServer.Connect(ProgID, NodeName)

Set MyGroups = AnOPCServer.OPCGroups

Set OneGroup = MyGroups.Add("AnOPCGroupName" ) Set AnOPCItemCollection = OneGroup.OPCItems

属性：Count

|  |  |
| --- | --- |
| 说明 | 只读的属性，返回OPC标签集合中的项数。 |
| 文法 | Count As Long |
| 示例 | Dim CurrentValue As Long  ‘ 读取OPC标签的数目  CurrentValue = AnOPCItemCollection.Count |

方法：AddItems

|  |  |
| --- | --- |
| 说明 | 在OPC标签集合中添加新的OPC标签。其初期属性取决与OPC标签集合的默  认值。 |
| 文法 | AddItems (Count As Long, ItemIDs() As String, ClientHandles() As Long, \_ ByRef ServerHandles() As Long, ByRef Errors() As Long, \_  Optional RequestedDataTypes As Variant, \_ Optional AccessPaths As Variant)  Count 添加的OPC标签的数目。  ItemIDs 要添加的OPC标签的标识符的数组。 ClientHandles 要添加的OPC标签的客户句柄的数组。 ServerHandles 返回的对应添加的OPC标签的服务器句柄的数组。 Errors 返回的对应添加的OPC标签的错误码的数组。  RequestedDataTypes 选用参数，是要添加的OPC标签的要求的数据类型的数 组。  AccessPaths 选用参数，要添加的OPC标签路径的数组。 |
| 示例 | Dim ClientHandles(100) As Long Dim AnOPCItemIDs(100) As String Dim AddItemCount As Long |

Dim I As Long

AddItemCount = 10

‘ 设置要添加的OPC标签的客户句柄和项标识符

For I = 1 To AddItemCount

ClientHandles(I) = I + 1

AnOPCItemID(I) = "Item\_" & I Next

‘ OPC标签的添加

AnOPCItemCollection.AddItems AddItemCount, AnOPCItemIDs, \_ ClientHandles, AnOPCItemServerHandles, AnOPCItemServerErrors

方法：Remove

|  |  |
| --- | --- |
| 说明 | 清除指定的OPC标签。 |
| 文法 | Remove (Count As Long, ServerHandles() As Long, \_ ByRef Errors() As Long)  Count 要清除的OPC标签的数目。  ServerHandles 要清除的OPC标签的服务器句柄的数组。  Errors 返回的对应被清除OPC标签的错误码的数组。 |
| 示例 | ‘ 清除OPC标签  AnOPCItemCollection.Remove (AnOPCItemServerHandles, \_ AnOPCItemServerErrors) |

所有OPC标签集合支持的属性和方法的一览如表1-18和表1-19所示。详细资料请参照

6.5参考资料的“OPC数据访问自动化接口标准”。

表 *1-18 OPC*标签集合的属性

|  |  |  |
| --- | --- | --- |
| 属性名 | 属性 | 说明 |
| Parent | 只读 | 返回所属的OPC组对象。 |
| DefaultRequestedDataType | 读写 | 在添加OPC标签时，默认的要求数据类型。初  期值是VT\_Empty（＝控制设备的固有数据类 型）。 |
| DefaltAccessPath | 读写 | 在添加OPC标签时，默认的数据访问路径。初  期值是""（＝无路径）。 |
| DefaultIsActive | 读写 | 在添加OPC标签时，默认的活动状态。初期值  是True（真＝活动）。 |
| Count | 只读 | 集合对象的固有属性。OPC标签集合中的OPC  标签数。 |

表 *1-19 OPC*标签集合的方法

|  |  |
| --- | --- |
| 方法名 | 说明 |
| Item | 返回OPC标签集合中由集合索引（ItemSpecifier ）指定  的OPC标签。 |
| GetOPCItem | 返回OPC标签集合中由服务器句柄指定的OPC标签。 |
| AddItem | 在OPC标签集合中添加新的OPC标签。 |
| Remove | 清除指定的OPC标签。 |
| Validate | 检查被添加的OPC标签。 |

|  |  |
| --- | --- |
| SetActive | 分别设置OPC标签为活动状态或非活动状态。 |
| SetClientHandles | 设置OPC标签的客户句柄。 |
| SetDataTypes | 设置OPC标签的要求的数据类型。 |

**1.4.8 OPC**标签对象

OPC标签对象OPCItem表示与OPC服务器内某个数据的连接。各个OPC标签由数 据值，质量标志以及采样时间构成。所有OPC标签支持的属性和方法的一览如表1-20 和表1-21所示。详细资料请参照6.5参考资料的“OPC数据访问自动化接口标准”。

表 *1-20 OPC*标签的属性

|  |  |  |
| --- | --- | --- |
| 属性名 | 属性 | 说明 |
| Parent | 只读 | 返回所属的OPC组对象。 |
| ClientHandle | 读写 | 客户句柄是由应用程序指定的用于识别某个  OPC标签的长整型数。当OPC组事件发生时，服 务器将这个客户句柄和结果一起返回给OPC应 用程序。 |
| ServerHandle | 只读 | 服务器句柄是由OPC服务器设置的用于识别某  个OPC标签的一个独特长整型数。 |
| AccessPath | 只读 | 返回OPC应用程序指定的访问路径。 |
| AccessRight | 只读 | 返回OPC标签的访问权限。 |
| ItemID | 只读 | 返回识别这个OPC标签的标识符。 |
| IsActive | 读写 | 用以控制OPC标签的活动状态。只有活动状态的  OPC标签才进行定期的数据更新。 |
| RequestedDataType | 读写 | 要求的数据类型。 |
| Value | 只读 | 返回从OPC服务器读取的最新数据值。 |
| Quality | 只读 | 返回从OPC服务器读取的最新数据的质量标志。 |
| TimeStamp | 只读 | 返回从OPC服务器读取的最新数据的采样时间。 |
| CanonicalDataType | 只读 | 返回OPC服务器内固有的数据类型。 |
| EUType | 只读 | 返回工程单位（Engineering Unit）的数据类型。 |
| EUInfo | 只读 | 返回表示工程单位（Engineering Unit）的Variant  型数值。 |

表 *1-21 OPC*标签的方法

|  |  |
| --- | --- |
| 方法名 | 说明 |
| Read | 从OPC服务器中读取OPC标签的数据。 |
| Write | 向OPC服务器中写入OPC标签的数据。 |

**1.4.9 OPC**浏览器对象

OPC浏览器对象OPCBrowser是OPC服务器名称空间的枝和叶（项）的集合。浏览 功能是选用功能，OPC服务器不支持浏览的时候，即使执行CreateBrowser也不生成这 个对象。所有OPC浏览器支持的属性和方法的一览如表1-22和表1-23所示。详细资料 请参照6.5参考资料的“OPC数据访问自动化接口标准”。

表 *1-22 OPC*浏览器的属性

|  |  |  |
| --- | --- | --- |
| 属性名 | 属性 | 说明 |
| Organization | 只读 | OPC服务器的名称空间的类型（平层型或分层 |

|  |  |  |
| --- | --- | --- |
|  |  | 型）。 |
| Filter | 读写 | 使用ShowBranches 或ShowLeafs方法时的浏览对  象过滤器。使用这个过滤器可以缩小被浏览的名 称范围。 |
| DataType | 读写 | 使用ShowLeafs方法时，希望浏览的项的数据类  型。 |
| AccessRights | 读写 | 使用ShowLeafs方法时，希望浏览的项的访问权  限。 |
| CurrentPosition | 只读 | 返回在名称空间中的现在位置。 |
| Count | 只读 | 浏览结果中的浏览项数。 |

表 *1-23 OPC*浏览器方法

|  |  |
| --- | --- |
| 方法名 | 说明 |
| Item | 返回浏览结果中按集合索引（ItemSpecifier ）指定的对象。 |
| ShowBranches | 将现在位置下的所有符合过滤条件的枝（Branch）加入  到浏览结果中去。 |
| ShowLeafs | 将现在位置下的所有符合过滤条件的叶（Leav）加入到  浏览结果中去。 |
| MoveUp | 向现在位置的上一层移动。 |
| MoveToRoot | 向名称空间的最上层移动。 |
| MoveDown | 向现在位置的下一层移动。 |
| MoveTo | 向浏览器的绝对位置移动。 |
| GetItemID | 由浏览项的名称返回OPC标签的标识符。 |
| GetAccessPath | 返回OPC标签标识符的路径字符串。 |